The present invention relates to document transfer systems and in particular relates to such systems involving cryptographic protocols enabling a document to be obtained by a consumer upon payment to the owner of the document.

It would be desirable for the protocol to have strong fairness properties, i.e. a guarantee that, at the end of the protocol, either both the owner and the consumer receive payment and the document respectively, or neither party receives anything useful.

There is a substantial body of work on fair exchange and cryptographic services which use this primitive. For protocols requiring fairness in the absence of third parties, the definition of fairness is necessarily probabilistic, and such protocols are usually based on the gradual release of secrets. The following documents describe recent work on practical proposals for fair exchange which use a third party with varying trust assumptions:

(a) Matthew K. Franklin and Michael K. Reiter, *Fair Exchange with a Semi-Trusted Third Party*, Proceedings of the 4th ACM Conference on Computer and Communications Security, 1997; this document describes a fair exchange protocol with a semi-trusted third party with trust assumptions similar to those used in the present invention. The third party inthis case, however, is online even if the parties follow the protocol faithfully;

(b) US patent 5,666,420 entitled "Simultaneous Electronic Transactions" in the name of Silvio Micali describes an optimistic protocol for certified electronic mail with sleeping post offices;

(c) N. Asokan, M. Schunter and M. Waidner, *Optimistic Protocols for Fair Exchange*, Proceedings of the 4th ACM Conference on Computer and Communications Security, 1997; this document describes a practical optimistic protocol for fari exchange. However, this protocol increases the trust requirements on the third party in the event of a dispute resolution being required. In particular, the third party inspects the contents of a message containing the item being exchanged while resolving disputes. In addition, the described protocol family has a

synchronous time model which may not be suitable for certain applications.

In accordance with a first aspect of the present invention there is provided a cryptographic method of enabling a consumer to obtain a document from an owner upon a payment as defined in Claim 1.

In accordance with a second aspect of the present invention there is provided a cryptographic method of enabling a consumer to obtain a document from an owner upon a payment as defined by Claim 2.

The first and third portions of the key are preferably different.

The method may be arranged for enabling a said consumer to receive a plurality of such documents, wherein said key is divided into different respective sets of portions for each document.

The document source is preferably a printer.

In the preferred embodiment, the ordering protocol is carried out in the presence of a mediator with minimal trust assumptions. The protocol is optimistic, in that the mediator remains off-line except in the case of dispute resolution. In addition, the mediator does not learn any information about the document, even in the event of a dispute.

In accordance with a third aspect of the present invention there is provided a document source for use in one of the above-described methods as defined by Claim 7.

In accordance with a fourth aspect of the present invention, there is provided a document source as defined by Claim 8.

The document source is preferably a printer which is advantageously arranged to print a number of copies of a said document in each of a plurality of formats.

The printer may be arranged to print only one copy of a said document in a first format and an unlimited number of copies of said document in a second format.

The formats may comprise different resolutions or a choice of monochrome and colour images.

In accordance with a fifth aspect of the present invention, there is provided a fair exchange method of enabling a consumer to obtain a document from an owner upon a payment as defined by Claim 14.

In accordance with a sixth aspect of the present invention there is provided a cryptographic method of enabling a first party to obtain an item of value from a second party upon receipt by said second party of a second item of value as defined by Claim 15.

In accordance with a seventh aspect of the present invention there is provided a fair exchange method of enabling a contract between a buyer and a seller of a commodity as defined in Claim 16.

Fig. 1 shows in diagrammatic form the protocol of the preferred embodiment of the present invention.

Figure 2 shows in block diagram form a printer which implements the preferred embodiment of the present invention.

A preferred embodiment of the present invention will now be described with reference to the accompanying drawing which illustrates the protocol of the preferred embodiment.

The parties involved in the protocol are:

(a) The owner O of a printable document who wishes to charge end users for prints of the document. For the purposes of payment,

the copyright owner adopts the role of a merchant in SET terminology;

(b) The consumer C who wishes to print the copyrighted material and pay for it. C assumes the role of cardholder in SET terminology for the payment phase.

(c) The printer P which is the physical printer device intended to print the document. The printer can sign random nonces and perform ElGamal decryptions. It has its private signing key in a temper-resistant store. Its function upon receipt of a document is to run the protocol to recover the encryption key and thereafter to decrypt and render the document. The printer can understand a copyright specification language and is trusted by the owner to follow the agreement. The printer does not need to verify any signatures, this being performed by the consumer. There is thus no requirement that any set of trusted root keys be maintained;

(d) The mediator M, a semi-trusted third party who mediates the transaction between the owner and the consumer and who can arbitrate in the event that the parties do not follow the protocol.

As can be seen from the above, there is a distinction between the two roles of the consumer and the printer. This is important, because of the underlying trust assumption, namely that the printer can be entrusted by the owner to respect the conditions of a copyright agreement, whereas it may not be reasonable to assume that the consume would do the same.

The requirement for a payment protocol are that a payment request can be linked to some order information in a non-repudiable manner and that it is possible for the consumer to query the status of a payment request. For illustration, the SET protocol is used below to describe the payment phase.

The parties share a large Sophie Germain prime number p, i.e. one satisfying the relation:

$$q = (p - 1) / 2,$$

where q is also a prime number.

The documents are distributed in an encrypted manner using offline or online means. The (bulk) encryption uses key material derived from an integer $x \in \mathbf{F}_p$. This can be done, for example, by computing a one-way transformation $H(x)$ and using some output bits as the initialisation vector (IV) and bulk enciphering key k.

The protocol has the following properties:

- The secret corresponding to a document is not revealed to the consumer.
- If the owner deviates from the protocol, the consumer has an undeniable proof linking the owner to the (incomplete) transaction. It can resolve the dispute with the mediator. This is the only trust assumption on the mediator – in case the owner deviates from the protocol, it promises to perform the dispute resolution steps faithfully.
- In the event of a secret associated with a document being invalid, the printer can provably demonstrate its invalidity.
- The mediator does not gain any knowledge of the secret unless one of the parties reveals the necessary information.
- The protocol is optimistic – if parties execute the protocol steps faithfully, mediation is not required.

The parties share parameters (g, h) where g generates a multiplicative group of prime order p. The element h generates a multiplicative group of prime order $q = (p - 1) / 2$. The group orders should be chosen such that the discrete logarithm problem is hard in the corresponding multiplicative groups. Computations, unless otherwise specified, occur in the finite field $\mathbf{F}_p$.

The parties choose secrets in $\mathbf{F}_q$ and publish corresponding public keys in $\mathbf{F}_p$. The mediator's public key is:

$$y_m = h^{s_m},$$

and the owner's public key is:

$$y_0 = h^{s_0}.$$

The printer and owner use a signature mechanism σ such that the printer's signature σ_p and σ_0 are verifiable by the owner, the consumer and the mediator.

A public commitment $C(x) - g^x \bmod p$ to the bulk encryption key is attached to the encrypted content and signed by the owner. We use a protocol described in a paper by Markus Stadler, entitled *Publicly Verifiable Secret Sharing* (Advances in Cryptology – EUROCRYPT '96, Lecture Notes in Computer Science, **1070** (1996), 190 – 199) to publicly verify the link between a commitment of the form $g^x \bmod p$, and the ElGamal encryption of the associated secret x in a known public key using a computationally zero knowledge argument. This arrangement is shown in Figure 1.

Prior to ordering a document, the owner and consumer agree on a mediator acceptable to both parties. The parties also agree upon an acceptable copyright agreement specified in a manner that is understood by the printer. We denote this string by R.

The owner generates a nonce $n_0 \in Z / p Z$ and sends it to the consumer to initiate the protocol.

The consumer sends the document's published value g^x, the copyright string R and the owner's nonce n_0 to the printer. The printer checks if the string R specifies controls that it can perform. If this is the case, it generates a random number $r_p \in Z / q Z$, computes the nonce:

$$n_p = h^{r_p},$$

and sends the tuple (n_0, n_p, R) signed to the consumer. The nonce n_p will be treated by the printer as a one-time public key exchange key for this transaction. The printer

internally keeps track of the association between the copyright string, the public commitment and its own one-time key.

The consumer passes the printer's signed message on to the owner, in addition to any payment-related payload indicating initialisation (for SET, this will be a message to the merchant requesting a wake-up of the consumer's wallet software).

The owner, on receipt of the print request, generates a random value of w between 1 and $q - 1$. It now shares x between the printer and the mediator using a publicly-verifiable 2-out-of-2 sharing scheme.

Secret sharing is effected by performing an ElGamal encryption of w using the printer's one-time key exchange key n_p and of $x - w$ in the mediator's public key y_m.

The owner generates ElGamal tuples:

$$X_p = (h^r, n^r{}_p / w) \quad \text{and} \quad X_m = (h^s, y^s{}_m / (x - w))$$

for some values of r, s chosen uniformly at random from $[1, \ldots, q - 1]$.

The values g^w and g^{x-w}, encrypted tuples X_m and X_p and tuple (n_0, n_p, R) are now sent signed to the consumer in addition to any payment-related payload (for SET, this would be the wake-up message to the consumer's wallet and would include the order data component from the owner).

The consumer (without input from the printer) can verify that the sharing was correctly performed by the owner, and that the encryption is valid. This is effected using the protocol defined in the above-mentioned paper by Stadler.

The consumer now generates its payment message signifying an intended purchase for the key material x. The payment request is cryptographically linked with the nonces associated with the transaction. In case of SET, the transaction would be linked to the Order Data element in the protocol message.

Upon validation of the payment request, the owner now sends the other half of the bulk decryption key encrypted in the one-time key exchange key n_p; that is, it sends $X'_p = (h^t, n^t_p / (x - w))$ to the consumer. This is bound with the nonce pair and signed by the owner.

The printer, once in possession of the ElGamal pairs X_p and X'_p recovers x using its one-time secret r_p. It can now decrypt the document using x to derive the bulk decipherment key. The printer then proceeds to print the document in agreement with the associated copyright agreement R.

The above protocol is shown diagrammatically in Figure 1.

The optimistic version of the protocol favours the copyright owner, because the consumer pays for the goods before the encryption key is released by the owner.

In case of disputes, the consumer can present the public transaction details (n_0, n_p, R), the encrypted shares X_p and X_m and the payment information to the mediator. The mediator can verify the cryptographic link between the payment and the mediator's share. Once this is done, she queries the owner with the transaction ID of the payment. Upon receipt of a satisfactory response (or no response at all), she extracts her share ($x - w$) and supplies an ElGamal encryption of ($x - w$) in the appropriate one-time key exchange key of the printer. Since ElGamal encryption is a randomised construction, this will result in a different ElGamal pair with a probability approaching 1.

In addition, the SET 1.0 payment protocol requires resolution outside of the protocol if the merchant does not deliver an InqRes back to the mediator.

The following demonstrates that the protocol is l-resilient, i.e. any one party can deviate from the protocol without compromising its security, under the decision Diffie-Hellman assumption. The decision Diffie-Hellman assumption is described in a paper by Stefan Brands entitled *An efficient off-line electronic cash scheme based on the representation problem* (Technical Report CS-R9323, CWI, Amsterdam, 1993):

- The moderator only knows $(x - w)$ and hence does not learn anything about the secret x unless one of the parties discloses it. This holds even if the moderator resolves a dispute successfully.

- If the consumer deviates by not providing a valid payment message, the owner aborts the protocol. Since the secret sharing is perfect zero knowledge, no information about x is revealed.

- If the owner does not provide the final encryption X'_p of $(x - w)$, the consumer can request the moderator to decrypt its share X_m. The moderator can now compute $(x - w)$ and send it to the consumer encrypted in the printer's one-time key exchange key n_p.

- The consumer cannot reply previously collected printer tokens (ElGamal signatures by the owner) to the printer, because the key exchange key for the printer, n_p, will be different from a previous transaction with a probability approaching 1.

- An arbitrary party can build shares corresponding to some x', but it is the consumer's responsibility to tie the commitment $g^{x'}$ of x' to the owner's identity by verifying the owner's signature.

The zero-knowledge protocol described in the above-mentioned paper by Stadler has a round efficiency of ½ and hence requires a significant number of rounds to reduce the error probability acceptably. As noted in Stadler, this proof can be made non-

interactive. As a result, the proof need not be performed online by the consumer. Instead, the consumer only checks if the public commitments $C_1 = g^w$ and $C_2 = g^{x-w}$ match the document's public value $C = g^x$, i.e. whether the relation $C_1C_2 = C$ holds true.

In case of a dispute, the mediator can decrypt and verify her share (x-w). She can now run the proof for the printer's share X_p. As a result, the protocol is optimistic but fair.

A printer for use in the above methods is described with reference to Figure 2. The printer 1 comprises a document memory 2 for storing a received encrypted document, a key memory 3 for storing a received first cryptographic key portion from the owner, a processor 4 for receiving a second cryptographic key portion from the owner and combining it with the first key portion stored in the key memory 3 to form a complete cryptographic key which is supplied to a decrypting module 5. The encrypted document stored in the document memory 2 is supplied to the decrypting module 5 whereupon the document is decrypted and supplied to the consumer.